



DEPARTMENT OF AIML
23CST202- OPERATING SYSTEMS
II YEAR IV SEM AIML-B
UNIT 3-MEMORY MANAGEMENT
TOPIC –ACCESS METHODS

ACCESS METHODS

File access methods in an operating system are the techniques and processes used to read from and write to files stored on a computer's storage devices. There are several ways to access this information in the file. Some systems provide only one access method for files. Other systems, such as those of IBM, support many access methods, and choosing the right one for a particular application is a major design problem.

These methods determine how data is organized, retrieved, and modified within a file system. Understanding file access methods is crucial for efficient data management and system performance. In this article, we are going to discuss different types of methods to access the file.

There are three ways to access a file in a computer system:

- Sequential-Access
- Direct Access
- Index sequential Method

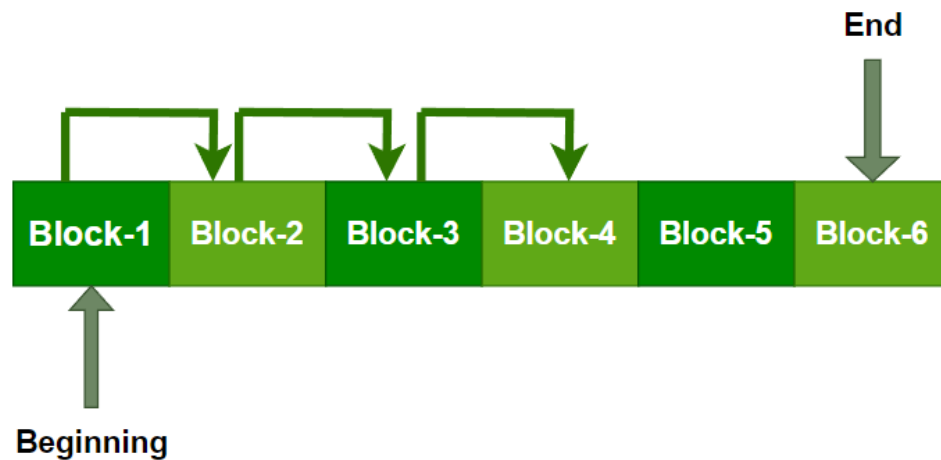
Sequential Access

It is the simplest access method. Information in the file is processed in order, one record after the other. This mode of access is by far the most common; for example, the editor and compiler usually access the file in this fashion.

Read and write make up the bulk of the operation on a file. A read operation -read next- reads the next position of the file and automatically advances a file pointer, which keeps track of the I/O location. Similarly, for the -write next- append to the end of the file and advance to the newly written material.



Sequential Access Method



Sequential Access Method

Key Points related to Sequential Access

- Data is accessed from one record right after another record in an order.
- When we use the read command, it moves ahead pointer by one.
- When we use the write command, it will allocate memory and move the pointer to the end of the file.
- Such a method is reasonable for tape.

Advantages of Sequential Access Method

- It is simple to implement this file access mechanism.
- It uses lexicographic order to quickly access the next entry.
- It is suitable for applications that require access to all records in a file, in a specific order.
- It is less prone to data corruption as the data is written sequentially and not randomly.
- It is a more efficient method for reading large files, as it only reads the required data and does not waste time reading unnecessary data.
- It is a reliable method for backup and restore operations, as the data is stored sequentially and can be easily restored if required.

Disadvantages of Sequential Access Method

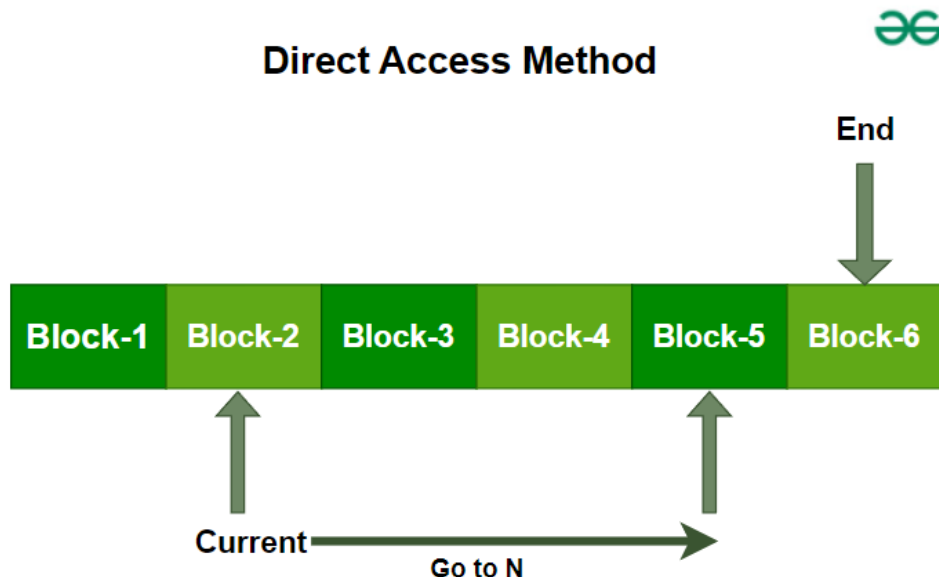
- If the file record that needs to be accessed next is not present next to the current record, this type of file access method is slow.
- Moving a sizable chunk of the file may be necessary to insert a new record.
- It does not allow for quick access to specific records in the file. The entire file must be searched sequentially to find a specific record, which can be time-consuming.



- It is not well-suited for applications that require frequent updates or modifications to the file. Updating or inserting a record in the middle of a large file can be a slow and cumbersome process.
- Sequential access can also result in wasted storage space if records are of varying lengths. The space between records cannot be used by other records, which can result in inefficient use of storage.

Direct Access Method

The **Direct Access Method** (also known as the **Relative Access Method**) allows rapid reading and writing of fixed-length records in any order. Based on the disk model, which supports random access, the file is viewed as a sequence of numbered blocks or records. This means data can be read or written to any block in no specific order. The block numbers provided by the user are relative, starting from 0, 1, and so on. There are no restrictions on the order of accessing records in direct access files.



Direct Access Method

Advantages of Direct Access Method

- The files can be immediately accessed decreasing the average access time.
- In the direct access method, in order to access a block, there is no need of traversing all the blocks present before it.

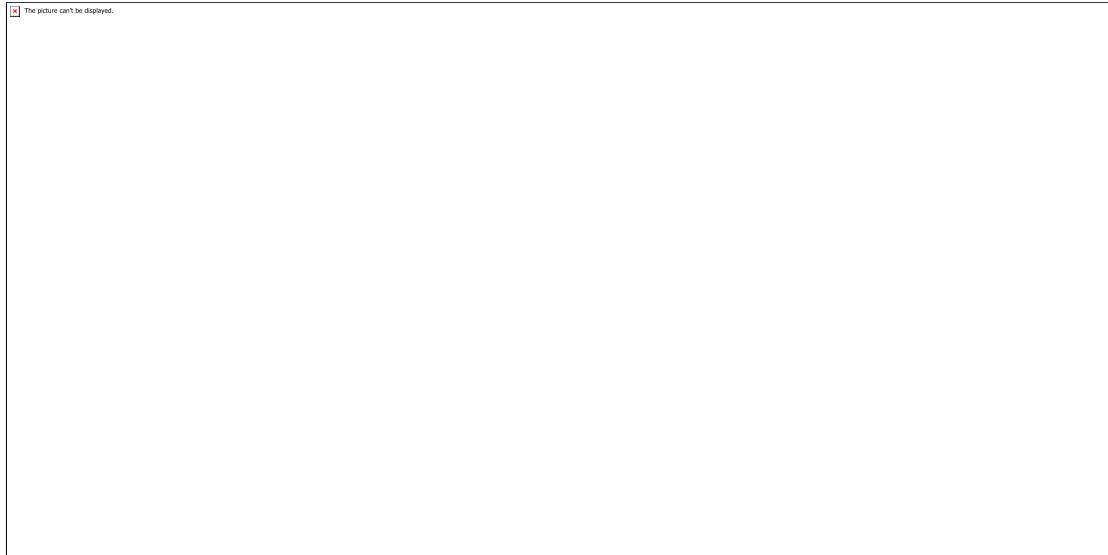
Disadvantages of Direct Access Method

- **Complex Implementation** : Implementing direct access can be complex, requiring sophisticated algorithms and data structures to manage and locate records efficiently.
- **Higher Storage Overhead** : Direct access methods often require additional storage for maintaining data location information (such as pointers or address tables), which can increase the overall storage requirements.



Index Sequential method

It is the other method of accessing a file that is built on the top of the sequential access method. These methods construct an index for the file. The index, like an index in the back of a book, contains the pointer to the various blocks. To find a record in the file, we first search the index, and then by the help of pointer we access the file directly.



Index Access Method

Key Points Related to Index Sequential Method

- It is built on top of Sequential access.
- It control the pointer by using index.

Advantages of Index Sequential Method

- **Efficient Searching** : Index sequential method allows for quick searches through the index.
- **Balanced Performance** : It combines the simplicity of sequential access with the speed of direct access, offering a balanced approach that can handle various types of data access needs efficiently.
- **Flexibility** : This method allows both sequential and random access to data, making it versatile for different types of applications, such as batch processing and real-time querying.
- **Improved Data Management** : Indexing helps in better organization and management of data. It makes data retrieval faster and more efficient, especially in large databases.
- **Reduced Access Time** : By using an index to directly locate data blocks, the time spent searching for data within large datasets is significantly reduced.

Disadvantages of Index Sequential Method



- **Complex Implementation** : The index sequential method is more complex to implement and maintain compared to simple sequential access methods.
- **Additional Storage** : Indexes require additional storage space, which can be significant for large datasets. This extra space can sometimes offset the benefits of faster access.
- **Update Overhead** : Updating the data can be more time-consuming because both the data and the indexes need to be updated. This can lead to increased processing time for insertions, deletions, and modifications.
- **Index Maintenance** : Keeping the index up to date requires regular maintenance, especially in dynamic environments where data changes frequently. This can add to the system's overhead.

Other Way of File Access

1. Relative Record Access

Relative record access is a file access method used in operating systems where records are accessed relative to the current position of the file pointer. In this method, records are located based on their position relative to the current record, rather than by a specific address or key value.

Key Points Related to Relative Record Access

- Relative record access is a random access method that allows records to be accessed based on their position relative to the current record.
- This method is efficient for accessing individual records but may not be suitable for files that require frequent updates or random access to specific records.
- Relative record access requires fixed-length records and may not be flexible enough for some applications.
- This method is useful for processing records in a specific order or for files that are accessed sequentially.

Advantages of Relative Record Access

- **Random Access:** Relative record access allows random access to records in a file. The system can access any record at a specific offset from the current position of the file pointer.
- **Efficient Retrieval:** Since the system only needs to read the current record and any records that need to be skipped, relative record access is more efficient than sequential access for accessing individual records.
- **Useful for Sequential Processing:** Relative record access is useful for processing records in a specific order. For example, if the records are sorted in a specific order, the system can access the next or previous record relative to the current position of the file pointer.

Disadvantages of Relative Record Access



- **Fixed Record Length:** Relative record access requires fixed-length records. If the records are of varying length, it may be necessary to use padding to ensure that each record is the same length.
- **Limited Flexibility:** Relative record access is not very flexible. It is difficult to insert or delete records in the middle of a file without disrupting the relative positions of other records.
- **Limited Application:** Relative record access is best suited for files that are accessed sequentially or with some regularity, but it may not be appropriate for files that are frequently updated or require random access to specific records.

2. Content Addressable Access

Content-addressable access (CAA) is a file access method used in operating systems that allows records or blocks to be accessed based on their content rather than their address. In this method, a hash function is used to calculate a unique key for each record or block, and the system can access any record or block by specifying its key.

Keys in Content-Addressable Access

- **Unique:** Each record or block has a unique key that is generated using a hash function.
- **Calculated based on content:** The key is calculated based on the content of the record or block, rather than its location or address.

Key Points Related to Content-Addressable Access

- Content-addressable access is a file access method that allows records or blocks to be accessed based on their content rather than their address.
- CAA uses a hash function to generate a unique key for each record or block.
- CAA is efficient for searching large databases or file systems and is more flexible than other access methods.
- CAA requires additional overhead for calculating the hash function and may have collisions or limited key space.

Advantages of Content-Addressable Access

- **Efficient Search:** CAA is ideal for searching large databases or file systems because it allows for efficient searching based on the content of the records or blocks.
- **Flexibility:** CAA is more flexible than other access methods because it allows for easy insertion and deletion of records or blocks.
- **Data Integrity :** CAA ensures data integrity because each record or block has a unique key that is generated based on its content.

Disadvantages of Content-Addressable Access



- **Overhead:** CAA requires additional overhead because the hash function must be calculated for each record or block.
- **Collision:** There is a possibility of collision where two records or blocks can have the same key. This can be minimized by using a good hash function, but it cannot be completely eliminated.
- **Limited Key Space:** The key space is limited by the size of the hash function used, which can lead to collisions and other issues.